Integration d'un correcteur orthographique dans l'editeur structure GRIF

Patrice Frison, Eric Picheral, Hélène Richy

INRIA

UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1566

Programme 1
Architectures parallèles, Bases de données,
Réseaux et Systèmes distribués

INTÉGRATION D'UN CORRECTEUR ORTHOGRAPHIQUE DANS L'ÉDITEUR STRUCTURÉ GRIF

Patrice FRISON
Eric PICHERAL
Hélène RICHY

Décembre 1991

RR-1566

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTEMES ALEATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CEDEX FRANCE
Tél. : 99 84 71 00 - Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

Intégration d'un correcteur orthographique dans l'éditeur structuré Grif

Patrice Frison†, Eric Picheral‡ et Hélène Richy†
(Programme 1) **(Cicb)** **(Atelier)**

Publication Interne n°609 - Octobre 1991 - 22 pages

† *Irisa, campus de Beaulieu, F-35042 Rennes cedex, France*
‡ *CICB, campus de Beaulieu, F-35042 Rennes cedex, France*

Résumé

Nous présentons dans cet article l'intégration d'un correcteur orthographique dans l'éditeur de documents structurés Grif. Ce correcteur, basé sur l'utilisation de dictionnaires, détecte des fautes dans les parties textuelles d'un document édité par Grif et fait instantanément des suggestions de correction pertinentes. Nous montrons que cette méthode peut être implémentée efficacement et produire de bon résultats avec un temps de réponse réduit sur une station de travail d'architecture RISC en utilisant de grands dictionnaires. L'algorithme utilisé étant indépendant de la langue, ce correcteur peut traiter aussi bien des textes écrits en français que des textes écrits en anglais ou des textes multilingues grâce à l'utilisation d'un attribut *Langue* dans les documents Grif.

Multilingual string-to-string correction in Grif

Abstract

This paper describes the integration of a spelling corrector into the structured editor Grif. This corrector is based on the Levenshtein metric concept which is particularly efficient for string correction. This method can be implemented efficiently and can produce good results with short response time on a new RISC workstation even with large dictionaries. The integration within Grif enables checking of textual content of structured documents where large vocabularies are required. Thanks to attribute *Language* the editor can automatically adapt the correction to the language and can apply specific word recognition algorithm and dictionaries, thus allowing checking and correcting of multilingual documents.

Mots-clé

correcteur orthographique, multilinguisme, dictionnaires, éditeur.

1 Introduction

Nous présentons dans cet article l'intégration d'un correcteur orthographique dans l'éditeur de documents structurés Grif.

1.1 Grif: un éditeur de documents structurés

Grif est un éditeur de documents structurés. Il travaille à partir de structures génériques et de schémas de présentations. Grâce aux structures génériques, les documents produits sont toujours "bien" structurés. Grâce aux schémas de présentation, les documents sont "bien" présentés.

Ainsi, en utilisant les structures et les schémas de Grif, l'auteur d'un Article, par exemple, est-il sûr que la bibliographie sera bien présentée.

Les nouveaux développements entrepris autour de Grif dans le projet Opéra ont pour objectif de tirer profit de cette structuration forte pour améliorer encore le contenu des documents: identifier la langue de certains éléments permet de respecter les usages linguistiques et typographiques de chaque langue; identifier les passages à indexer permet de construire automatiquement des index; généraliser les liens existants entre documents permet de construire des hypertextes, etc. Il nous a paru intéressant tout d'abord d'intégrer à Grif [20] un correcteur orthographique qui soit capable de faire, dans un délais très court, des propositions de correction pertinentes.

1.2 Vérification et correction orthographique

Le contenu des documents produits par les systèmes de PAO présente souvent de nombreuses erreurs. Les principaux types d'erreurs que l'on rencontre dans les textes produits à l'aide de tels systèmes sont les suivants:

- les fautes de frappe: doublement de certaines lettres, inversion de lettres (dyslexie), frappe d'une touche voisine sur le clavier, confusion entre les touches de clavier différents QWERTY ou AZERTY (pour les usagers français), oubli d'une lettre,
- les erreurs d'édition (couper/coller): répétition d'un mot ou même d'une phrase entière, absence d'un mot, mots collés, coupures de mots,

- les fautes d'usage dues généralement à un manque de concordance entre l'écrit et l'oral, les fautes d'orthographe sur des mots peu usités du vocabulaire, la confusion entre orthographes différentes dans deux langues voisines, les confusions d'accent, etc.

- les fautes de grammaire (accords, conjugaisons),

- le non respect des règles typographiques: les espacements autour des signes de ponctuation, les abréviations, la typographie des sigles, l'usage des capitales initiales, les règles de césure, etc.

Pourtant, s'il n'existe pas d'outil d'aide permettant de vérifier que toutes les règles typographiques[1] sont respectées, la plupart des traitements de texte actuellement commercialisés comportent une fonction de correction orthographique intégrée. Mais, généralement les propositions de correction sont peu pertinentes et/ou longues à obtenir.

En plus du vocabulaire général de la langue, les textes contiennent souvent des noms propres et des sigles dont l'orthographe doit pouvoir être contrôlée. Les documents électroniques contiennent également des codes internes, des commandes d'édition (changement de police, commandes de mise en page, références numérotées, etc.) qui ne doivent pas être confondues avec le vocabulaire de la langue (par exemple, *documentstyle* est une commande pour LaTeX, mais c'est un mot incorrect en anglais). Il est donc important dans un document électronique de pouvoir différencier ces commandes des mots du texte afin de mieux les corriger.

Dans cet article, nous nous intéressons uniquement à la correction orthographique du contenu textuel des documents et en particulier à l'intégration d'un correcteur dans un éditeur structuré tel que Grif. La section 2 décrit la méthode de correction qui est intégrée dans Grif[2] en insistant particulièrement sur les optimisations réalisées et la mise en œuvre. Cette méthode, basée sur l'utilisation de dictionnaires, permet de corriger les erreurs de frappe et les fautes d'orthographe (sur les mots). Elle ne tient pas compte des règles

[1] Les ouvrages [5], [10], [11] servent de référence dans le domaine de la typographie: ils décrivent les règles communément admises pour l'usage de lettres capitales, de la ponctuation, de l'italique, des abréviations, des unités de mesure, la coupure des mots ou encore la présentation des notes et des références bibliographiques.

[2] Cette méthode avait été précédemment expérimentée à l'Irisa dans le projet API sur une machine systolique [8].

de grammaire. En conséquence, les fautes d'accord ne seront pas détectées. Cependant, les travaux menés sur le sujet [16][18] confirment que 80% des erreurs sur les mots sont des erreurs purement dactylographiques (dues à des substitutions, omissions, permutations ou insertions de caractères). La section 3 décrit les problèmes posés par l'intégration d'un correcteur orthographique dans Grif en insistant sur les aspects spécifiques à la langue, la reconnaissance des mots et l'utilisation des dictionnaires.

2 Méthode de correction

Dans le domaine qui nous intéresse, *la correction de textes écrits en langue naturelle*, de nouvelles méthodes de correction dotées d'une certaine capacité d'analyse syntaxique commencent à apparaître [17] ; les recherches basées sur l'analyse du langage naturel (linguistique et intelligence artificielle) devraient permettre de définir des méthodes permettant à la fois la détection des fautes grammaticales et leur correction (par exemple, sur l'accord des participes passés [21]). Mais, en l'absence de méthode de correction automatique satisfaisante, il faut encore envisager actuellement l'intervention de l'utilisateur : dans ce cas, on parle de correction interactive de documents.

Un certain nombre de méthodes appelées souvent **méthodes statistiques** tiennent compte de la fréquence observée d'apparition de lettres dans les mots.

La *méthode des abréviations* consiste à associer à chaque mot son abréviation, constituée par certaines lettres de ce mot, compte tenu de la position de ces lettres dans le mot et de le fréquence d'apparition de ces lettres dans la langue anglaise [4]. Un mot erroné peut alors être corrigé pas les mots ayant même abréviation.

Dans *la méthode par décomposition*, un mot est décomposé en digrammes[3] et trigrammes ; le résultat de la comparaison de ces digrammes et trigrammes avec les di- et trigrammes d'un dictionnaire produit un indice de similitude permettant de désigner la chaîne la plus proche. Dans cette méthode, c'est le contenu du dictionnaire qui est le résultat d'études statistiques sur les digrammes et/ou trigrammes. Ce type de méthode permet surtout de corriger les erreurs de substitution. Une adaptation à la représentation *phonétique*

[3]Groupe de deux lettres représentant un seul son.

des mots permet de prendre en compte les fautes sur des trigrammes phonétiques [3].

La *méthode des alphacodes* consiste à associer à chaque mot son alphacode, c'est à dire la liste de ses lettres ordonnées par ordre alphabétique. En anglais, il y a peu d'anagrammes (en moyenne 1,10 mots ont même alphacode en anglais, 1,05 en français). Le système de correction consistera à utiliser deux dictionnaires : un dictionnaire des mots du vocabulaire et un dictionnaire des alphacodes et des mots associés. Un mot erroné (détecté à l'aide du premier dictionnaire) peut alors être corrigé par les mots ayant même alphacode ou par des mots ayant un alphacode voisin (une lettre en plus, en moins ou différente) et suffisamment proches du mot erroné.

D'autres méthodes ont été proposées : des méthodes combinatoires, des méthodes métriques et des méthodes combinant les précédentes.

Les **méthodes combinatoires** [15] consistent à générer toutes les chaînes possibles dont pourrait dériver le mot erroné en appliquant les opérations de substitution, permutation, insertion et omission.

L'intersection entre cet ensemble de chaînes et le dictionnaire fournit une ou plusieurs propositions de correction. Mais, pour éviter de manipuler de trop gros ensembles de chaînes, on se limite le plus souvent à une seule erreur par mot : ainsi, pour un mot de n caractères issus d'un alphabet de m lettres, il n'y aura que $m(2n+1) + n - 1$ chaînes à manipuler !

Les **méthodes métriques**, qui consistent à comparer un mot erroné avec les mots d'un dictionnaire et à calculer la *distance* entre ces mots, sont celles qui donnent actuellement les meilleurs résultats. Contrairement aux méthodes combinatoires, elles permettent de prendre en compte plusieurs erreurs par mot. Mais, comme elles sont très onéreuses en temps de calcul, elles sont rarement utilisées intégralement. Une extension des méthodes métriques au calcul des *distances phonétiques* a également été proposée par Véronis [23].

La méthode métrique de Levenshtein [12] est la plus connue : la distance entre deux mots M et M' est fonction du nombre d'opérations nécessaires à la transformation du mot M en M' ; elle a fait l'objet de nombreuses améliorations.

Nous décrivons ci-dessous la mise en œuvre de cette méthode dans l'éditeur structuré Grif ; elle a fait l'objet d'optimisations qui la rendent utilisable sur les postes de travail actuels (rapides) sans nuire à la qualité des résultats.

2.1 Algorithme de correction

La méthode métrique introduite par Levenshtein consiste à calculer une distance entre une chaîne erronée M et une chaîne M' d'un dictionnaire. Wagner et Fischer [24] ont défini cette distance, appelée **distance d'édition**, comme une fonction du nombre d'opérations d'édition nécessaires à la transformation de M en M'.

Les opérations d'édition qu'ils ont considéré sont les suivantes :

- la *substitution* : remplacement d'un caractère par un autre,
- l'*insertion* : ajout d'un caractère,
- l'*omission* : suppression d'un caractère.

La chaîne la plus proche de la chaîne erronée sera celle qui nécessite le moins d'opérations d'édition. En associant un coût à ces différentes opérations, on peut encore améliorer les résultats : la distance entre deux chaînes tiendra compte des probabilités d'insertion, d'omission ou de substitution des différents caractères ; en d'autres termes, la distance d'édition entre deux chaînes M et M' est le coût affecté à la suite d'opérations la moins coûteuse qui permet de transformer M en M'.

Plus formellement, soient deux chaînes à comparer,

M = (x_1,x_2,x_3,....x_n) et M' = (y_1,y_2,y_3,...,y_m),

soit d(x,y) le coût d'une opération changeant x par y, si Λ représente le caractère nul, d(x_i, Λ) représente le coût de la suppression du caractère x_i et d(Λ,y_j) représente le coût de l'insertion du caractère y_j,

Wagner a montré dans [24] que la distance δ(M, M') entre ces chaînes est obtenue par la relation de récurrence (1) :

$$(1) \quad D(i,j) = Min \begin{cases} D(i-1,j-1) + d(x_i,y_j) \\ D(i-1,j) + d(x_i,\Lambda) \\ D(i,j-1) + d(\Lambda,y_j) \end{cases}$$

avec les conditions initiales suivantes :

$$D(0,0) = 0,\ D(i,0) = D(i-1,0) + KO \text{ and } D(0,j) = D(0,j-1) + KI$$

KI et KO étant des constantes définissant respectivement le coût d'une opération d'insertion et le coût d'une opération d'omission.

Pour connaitre la distance entre deux chaînes de caractères M et M' de longueur respective n et m il faut donc calculer $n \times m$ valeurs. Ces valeurs constituent une matrice de dimension $n \times m$.

Par exemple, pour passer de la chaîne *allphait* à la chaîne *alphabet*, on voit sur le diagramme ci-dessous qu'il suffit des 3 opérations : 1 omission (de *l*), 1 insertion (de *b*) et 1 substitution (de *i* par *e*).

La distance entre ces deux chaînes peut être calculée à l'aide de la matrice ci-dessous (en supposant d(x,y) = 1 lorsque $x \neq y$) :

	a	l	l	p	h	a	i	t
a	**0**	1	2	3	4	5	6	7
l	1	**0**	**1**	2	3	4	5	6
p	2	1	1	**1**	2	3	4	5
h	3	2	2	2	**1**	2	3	4
a	4	3	3	3	2	**1**	2	3
b	5	4	4	4	3	**2**	2	3
e	6	5	5	5	4	3	**3**	3
t	7	6	6	6	5	4	4	**3**

Pour tenir compte de la *permutation* de caractères, la relation (1) a été étendue avec le terme suivant :

D (i - 2, j - 2) + γ (x_{i-1} x_i , y_{j-1} y_j) pour $i >= 2$ et $j >= 2$

avec

γ (x_{i-1} x_i , y_{j-1} y_j) = d (x_{i-1}, y_j) + d (x_i , y_{j-1}) + KP

KP étant une constante définie égale au coût de permutation.

2.2 Optimisations

Pour corriger un texte écrit en langue naturelle à l'aide d'un dictionnaire contenant tout le vocabulaire de cette langue, la mise en œuvre naïve de la méthode métrique nécessite de nombreux calculs: en effet, il faut calculer les distances entre chaque mot erroné et les mots de ce dictionnaire. Un dictionnaire anglais, par exemple, peut contenir plus de 200 000 mots de vocabulaire de base, auquel il faut ajouter les formes conjuguées (terminaison *-ed*, *-s* ou *-es*) ; un dictionnaire français, contient moins de vocabulaire de base, mais beaucoup plus de formes fléchies (de tous les verbes conjugués). Compte-tenu des vitesses de calcul des ordinateurs, le temps de réponse était, jusqu'à maintenant, peu satisfaisant. C'est la raison pour laquelle P. Frison et D. Lavenier [8] ont utilisé en 1988 une architecture systolique afin de paralléliser l'exécution de cet algorithme. Mais, le fort accroissement de la puissance des postes de travail rend aujourd'hui possible l'utilisation de cet algorithme sans recourir à une machine spécialisée.

De plus, un certain nombre d'optimisations proposées dans [22] permettent d'améliorer encore ces performances. Ces optimisations consistent, d'une part à limiter le nombre de mots effectivement comparés au mot erroné, d'autre part à réduire le nombre d'opérations nécessaires pour calculer la distance entre deux mots. Nous les résumons brièvement ci-dessous:

1. En supposant que le nombre des omissions et des insertions dans un mot est limité (ce qui est une hypothèse très réaliste), une chaîne de caractère de longueur n ne sera comparée qu'à des chaînes dont la longueur est proche de n. Un lexique réduit sera donc extrait du dictionnaire, la comparaison ne s'effectuant qu'avec des mots dont la longueur est comprise entre $(n - r)$ et $(n + r)$. r est appelée la largeur de recherche. Cette optimisation est particulièrement efficace pour les mots courts et les mots longs qui sont les moins nombreux dans le dictionnaire (donnant lieu à un lexique de petite taille). Il est à noter également qu'il n'y a pas de perte de qualité dans les propositions du fait de cette optimisation.

2. Comme le chemin optimum utilisé pour calculer la distance entre deux mots s'éloigne peu de la diagonale, il s'avère inutile de calculer tous les coûts intermédiaires dans la matrice décrite en 2.1, on se limitera aux diagonales (3 diagonales si $r = 1$ ou 5 diagonales si $r = 2$).

3. Il est inutile de retenir trop de mots : ainsi, les mots dont la distance avec le mot erronné dépasse un certain *seuil* ne seront pas retenus ; ce seuil dépend de la longueur du mot. Bien sûr, plus le mot incorrect est long, plus les possibilités de correction sont grandes. Mais l'usage du seuil permet de limiter les propositions aux mots les plus "proches" du mot erroné. En conséquence, le calcul de la matrice, qui s'effectue ligne par ligne, peut être stoppé dès que toutes les valeurs d'une ligne dépassent le seuil.

4. En triant les mots du dictionnaire par longueur et par ordre alphabétique, on peut éviter de refaire les calculs des préfixes communs à deux mots consécutifs de même longueur. Par exemple pour calculer la distance avec le mot *alterner* qui suit *alphabet* dans le dictionnaire, on conserve les deux premières lignes de la matrice. De plus, si le calcul du mot *alphabet* s'est arrêté à la deuxième ligne en raison d'un dépassement de seuil, il est inutile de calculer la distance pour le mot *alterner*.

5. Enfin, on ne retiendra finalement qu'un petit nombre de propositions (10 est un maximum réaliste). Les mots dont la distance n'est pas parmi les dix meilleures ne seront donc pas retenus : dès que le calcul de la distance dépasse la distance maximale d'un mot retenu, on peut passer au calcul de la distance suivante.

2.3 Mise en œuvre

Cette méthode peut être adaptée pour corriger un texte en langue naturelle selon que l'on s'intéresse plus particulièrement aux fautes de frappe (usage des claviers) ou aux erreurs phonétiques[4] (différence entre langue parlée et langue écrite).

Afin de mieux corriger les fautes de frappe, les coûts des opérations ont été définis en tenant compte de la position des touches sur le clavier. En outre, comme il existe plusieurs types de claviers [13], le changement de clavier suscite un certain nombre de fautes supplémentaires. Par exemple, certaines touches sont placées différemment sur le clavier dactylographique

[4]La généralisation décrite dans [8] permet de tenir compte des compressions de certains phonèmes (*au* en o, en français, par exemple).

français (sur les machines à écrire et postes de travail de bureautique) et sur les claviers des postes de travail des informaticiens :

Clavier anglais	QWERTYUIOP	ASDFGHJKL	ZXCVBNM
Clavier français	AZERTYUIOP	QSDFGHJKL	WXCVBN

Le coût d'une opération de substitution du caractère x par le caractère y est donc le suivant :

$$d(x,y) = \begin{cases} b & \text{si } x = y \\ m & \text{si } x \text{ et } y \text{ sont voisins sur le clavier} \\ f & \text{dans les autres cas.} \end{cases}$$

b, m et f étant des poids constants (généralement $b = 0 < m < f$).

Pour tenir compte de la spécificité des langues comportant des caractères accentués (albanais, allemand, danois, espagnol, finlandais, flamand, français, hongrois, etc.) nous avons affecté d'un coût m, les omissions d'accent, les inversions d'accent et les fautes d'usage des diphtongues. Par exemple, pour corriger un texte français, les substitutions des caractères suivants sont affectées du coût m : *(a,à), (â,a), (é,e), (è,e), (ê,e), (ë,e), (î,i), (ï,i), (ô,o), (ù,u), (û,u), (è,ê), (è,é), (é,ê), (ü,ë), (œ,o), (ç,c).*

Les valeurs affectées aux différents coûts des opérations, ainsi que le seuil de rejet et la largeur de recherche, permettent d'agir sur les résultats produits par le correcteur.

Les valeurs suivantes, données à titre indicatif, donnent de bons résultats pour corriger un texte ; elles peuvent, bien sûr, être modifiées pour être mieux adaptées à un utilisateur (compte-tenu des fautes les plus fréquentes) :

- les coûts d'insertion (KI) et d'omission (KO) sont égaux et constants, indépendants du caractère : d (x_i, Λ) = KO = 5 et d (Λ, y_j) = KI = 5
- le coût de permutation (KP) est de 4,
- les trois valeurs de substitution (b, m, f) sont respectivement 0, 3 et 5,
- une insertion ou omission est permise tous les groupes de 5 lettres (KG = 5), en conséquence la largeur de recherche r_i pour un mot de longueur i est calculée par la formule suivante : $r_i = 1 + ((i\text{-}1) / KG)$

- le seuil de rejet, Seuil[i], dépend de la longueur i du mot et des coefficients KI et KG : Seuil[i] = KI + KI * (i-1) / KG

Le nombre maximum de propositions, fixé par défaut à 5, peut être modifié (voir l'interface Grif), mais sa valeur ne doit pas dépasser 10 (au delà, le calcul des propositions est généralement sans intérêt pour l'utilisateur et augmente inutilement le temps de calcul).

Le choix de ces paramètres donne des résultats très satisfaisant, aussi bien en ce qui concerne la qualité des propositions de correction que le temps de réponse, comme le montrent les résultats des mesures décrites ci-dessous.

Pour chaque longueur de mots entre 6 et 17 lettres, 10 mots ont été choisis arbitrairement et des fautes ont été commises. Le correcteur a ensuite été lancé sur chaque groupe de 10 mots avec un dictionnaire français de 140 000 mots. Les résultats sont résumés dans le tableau ci-dessus.

Taille	r	seuil	mots	traités	rapport	lignes	l/mot	UC
6	2	10	26985	5330	19.7%	7660	1.44	0.6
7	2	11	46829	7737	16.5%	11176	1.44	0.9
8	2	12	65303	9204	14.1%	13571	1.47	1.2
9	2	13	82230	12579	15.3%	19851	1.58	1.8
10	2	14	88386	12998	14.7%	21820	1.68	1.8
11	3	15	88714	15944	18.0%	28929	1.81	3.0
12	3	16	75508	13250	17.5%	24886	1.88	2.7
13	3	17	68846	14424	20.9%	28927	2.01	2.9
14	3	18	43682	10341	23.7%	23382	2.26	2.5
15	3	19	32295	8762	27.1%	21168	2.42	2.0
16	4	20	28552	8802	30.8%	22859	2.60	2.4
17	4	21	15966	5507	34.5%	15738	2.86	1.9

Pour chaque *taille* de mot à tester, ce tableau donne les valeurs de la *largeur* de recherche *r* et du *seuil*. Il donne dans la colonne *mots* le nombre de mots du dictionnaire à comparer. La colonne *traités* indique le nombre de mots pour lesquels des calculs sont nécessaires. La colonne *rapport* montre le taux des mots effectivement calculés. On remarque que l'optimisation qui consiste à ne pas recalculer les premières lignes de la matrice pour les mots ayant même préfixe est particulièrement efficace puisqu'elle permet de rejeter un mot dont on sait que les calculs sur le préfixe dépassent la valeur de seuil.

Par exemple, sur les mots de 9 lettres seulement 15.3% des mots entre 7 et 9 lettres sont calculés. En d'autres termes, 84.7% des mots ont le même préfixe que le mot précédent dans le dictionnaire, préfixe suffisamment long pour autoriser un rejet.

La colonne *lignes* donne le nombre de lignes de matrices traitées et la colonne suivante indique le nombre moyen de lignes à traiter par mot. Ce nombre varie entre 1.4 et 2.9 ce qui est faible par rapport au nombre de lignes de la matrice (entre 6 et 17 respectivement).

La dernière colonne *UC* donne le temps moyen en seconde d'unité centrale sur une station SUN 4/60 pour la correction d'un mot de la taille indiquée. On remarque que le temps de correction est inférieur à 2 secondes pour les mots de moins de 10 lettres. Ce temps passe à environ 3 secondes pour les mots entre 11 et 14 lettres. Ce phénomène provient du fait que le paramètre r passe de 2 à 3 entre 10 et 11 lettres et que le seuil de rejet augmente en conséquence. De ce fait, plus de calculs sont effectués avant de rejeter un mot.

Le taux de correction obtenu sur le traitement de 120 mots est le suivant : 94 mots (78.3%) sont reconnus en première position, 113 mots (94.2%) ont une bonne correction dans les 5 premières positions et 7 mots (5.8%) ne sont pas corrigés. Ces résultats sont donnés à titre indicatif. En effet, il est clair que le taux de correction dépend fortement du nombre d'erreurs introduits dans chaque mot. Rappelons que dans cette expérimentation les erreurs ont été produites manuellement et ne proviennent pas de textes réellement saisis.

3 Intégration dans Grif

Grif est un système de production de documents principalement destiné à la documentation scientifique et technique et, d'une façon plus générale, aux domaines où l'on manipule des documents fortement structurés [9][19]. Il est disponible sur station de travail Unix et utilise le système de fenêtres X. A la différence des logiciels de PAO, Grif ne s'intéresse pas uniquement à la forme graphique des documents. Ce n'est pas non plus un traitement de texte, bien qu'il puisse effectuer les principales fonctions de ces systèmes. C'est un éditeur structural, dont les concepts de base se comparent plutôt à ceux d'un éditeur syntaxique. Les développements actuels autour de Grif nous conduisent à nous intéresser davantage, à l'Irisa, au contenu textuel

des documents (correction orthographique, multilinguisme). Le correcteur orthographique a donc été intégré à Grif pour améliorer le contenu des textes des document écrits avec Grif, qu'il s'agisse de textes écrits en français ou de textes écrits en anglais. Intégré à l'environnement Grif, ce correcteur réalise la détection, la correction et la substitution des mots erronés.

- *La détection* de fautes de frappe dans un document Grif: le correcteur peut détecter plusieurs erreurs dans un même mot (insertion, omission, permutation ou substitution de lettres).
- *La correction*: le correcteur fait des suggestions de correction classées dans un ordre pertinent, basé sur les types d'erreur les plus probables: le coût de chaque type d'erreur est défini dans un fichier de paramètres. L'utilisateur précise également le type de clavier qu'il utilise. Ainsi la première proposition est-elle souvent le meilleur choix.
- *La substitution*: intégré dans Grif, l'éditeur effectue la substitution du mot erroné par le mot choisi par l'utilisateur parmi les propositions du correcteur ou proposé directement par l'utilisateur.

3.1 Reconnaissance des mots

Avant de pouvoir détecter des fautes dans un texte, il faut être capable de reconnaître les mots de ce texte : un mot est une chaîne de caractères délimitée par des **séparateurs**. On appelle **alphabet** l'ensemble des caractères qui peuvent être contenus dans un mot : ainsi on distinguera l'alphabet des mots anglais de l'alphabet des autres langues latines. En effet, hormis quelques rares mots d'origine étrangère qui ont conservé leur typographie d'origine, l'anglais ne contient pas de caractères accentués. Afin de pouvoir représenter les caractères accentués de ces langues, Grif utilise le code Isolatin1 (sur 8 bits) [13].

Généralement, les séparateurs sont les signes de ponctuation, les espaces ou l'apostrophe. Mais, en français, par exemple, l'apostrophe ne doit pas toujours être considérée comme un séparateur. En effet, elle est d'un usage courant pour éviter l'hiatus entre les articles singuliers et les mots commençant par une voyelle (*l'aurore* et non *la aurore*, *d'une* et non *de une*) ; mais elle est encore employée dans certains mots comme *aujourd'hui* dans lesquels elle doit être considérée comme un lettre normale. *aujourd'hui* est donc considéré

comme un mot et doit figurer sous cette forme dans le dictionnaire (alors que *aujourd* et *hui* n'y figurent pas, pris séparément). En anglais par contre, l'apostrophe est toujours un signe d'élision (contraction des verbes ou cas possessif).

Le trait d'union, qui entre dans la composition des mot composés, ne doit pas toujours être considéré comme un séparateur : en français, certains mots composés n'ont pas de sens pris séparément (exemple : *rez-de-chaussée*). Par contre, le trait d'union qui suit certains préfixes (*anti-*) est un séparateur de mots : l'utilisation de tels préfixes permettant de composer des mots nouveaux ne figurant pas dans un dictionnaire (*anti-impérialiste*).

Dans Grif, l'algorithme de recherche des mots parcourt tous les éléments textuels contenus dans la structure du document, que ce soient des paragraphes de texte ou des titres de tableau. Seuls les mots de plus de deux lettres, ne comprenant que des caractères de l'alphabet préalablement défini sont retenus pour le correcteur. En conséquence, les mots contenant des chiffres ou des caractères spéciaux (*, $, %, etc.) ne seront pas considérés comme pouvant appartenir à une langue naturelle.

Le correcteur (et les dictionnaires) ne fait pas de différence entre majuscule et minuscule : tous les mots retenus sont convertis en minuscule avant d'être traités par le correcteur. Cependant le correcteur reconnaît 3 types de mots : (1) les mots écrits tout en majuscule, (2) les mots en minuscules contenant une capitale initiale, (3) tous les autres mots, supposés écrits avec des caractères minuscules. Cette classification est utile à l'interface pour proposer des corrections respectant la typographie d'origine.

Cette fonction de recherche de mot est particulièrement intéressante dans un document structuré ; elle peut être utilisée pour parcourir toutes les feuilles de texte contenues dans l'arbre du document (parcours linéaire du document), comme c'est le cas pour la correction orthographique d'un document ; mais elle peut être également utilisée pour vérifier la typographie de certains mots, en ne parcourant que certains éléments dans la structure du document : pour mettre une capitale initiale aux mots des titres, par exemple. Plus généralement, cette fonction est utile pour les applications qui traitent le contenu des documents structurés.

La *détection* des erreurs consiste simplement à comparer chaque mot contenu dans le texte d'un document Grif aux mots contenus dans un ou plusieurs dictionnaires (recherche dichotomique).

3.2 Dictionnaires

Tous les mots des dictionnaires sont en minuscule. Le contenu de chaque dictionnaire est trié de façon à faciliter le traitement par l'algorithme de correction : c'est-à-dire par longueur de mot et par ordre alphabétique. Le dictionnaire d'une langue devrait contenir tous les mots corrects qui peuvent être écrits dans cette langue ; cela signifie qu'il contient toutes les formes fléchies des mots (les pluriels, les féminins et surtout toutes les formes conjuguées).

Compte tenu de la variété des domaines dans lesquels Grif est utilisé, il est nécessaire de pouvoir utiliser non seulement un dictionnaire contenant le vocabulaire de base de la langue, mais également un ou plusieurs dictionnaires spécialisés contenant des termes techniques ou des sigles spécifiques au domaine d'application ou à l'environnement de travail. Pour corriger un texte dans une langue donnée, le correcteur de Grif utilise au maximum 4 dictionnaires :

- le *dictionnaire général* de cette langue (environ 150 000 mots en français, plus de 200 000 mots en anglais),
- un *dictionnaire spécialisé* de cette langue : contenant des termes techniques qui sont absents du dictionnaire général. Par exemple : systolique, hypertexte, indentation, hyperdocument, postscript, etc.
- un *dictionnaire de sigles* ou de noms propres (n'utilisant aucune langue en particulier), pouvant contenir des caractères spéciaux (c'est à dire hors alphabet). Par exemple : andré, ascii, cnrs, didot, france, frison, grif, inria, irisa, picheral, etc.
- un *dictionnaire privé associé au document*[5]. Par exemple : alphacode, digramme, levenshtein, trigramme, wysiwyg, etc.

Afin d'éviter des accès inutiles aux dictionnaires, il est possible, dans les documents multilingues édités avec Grif, de préciser quelle est la langue utilisée. Par exemple, un rapport de recherche écrit en français comporte généralement deux résumés, l'un écrit en français (*Résumé*), l'autre écrit en anglais (*Abstract*) qui porte un attribut précisant que la langue de correction

[5] Le dictionnaire privé peut éventuellement être réutilisé pour plusieurs documents.

est l'anglais. Ainsi le correcteur, utilisé avec des dictionnaires français ne corrigera pas cet élément du document. Inversement, le correcteur utilisant des dictionnaires anglais, ne cherchera des fautes que dans les parties du document portant l'attribut de langue anglaise.

Grâce au mécanisme des attributs, cette notion de langue peut être appliquée à n'importe quel élément d'un document, aussi bien un mot, un paragraphe qu'au document entier. Cette possibilité n'existe pas dans les éditeurs moins structurés. Dans FrameMaker [7], par exemple, cette notion de langue s'applique uniquement aux paragraphes. De plus, en utilisant les schémas de structure de Grif, des valeurs par défaut peuvent être données aux attributs portés par certains éléments (par exemple sur l'*Abstract* dans le schéma utilisé pour éditer les rapports de recherche).

Dans les applications travaillant sur le contenu textuel des documents, l'usage de dictionnaire est habituel. Le même dictionnaire est souvent utilisé pour la correction orthographique et pour la césure. Ces dictionnaires, contiennent des informations complémentaires sur la césure de chaque mot (par exemple, le dictionnaire de FrameMaker [7]), et sont utilisés par les algorithmes de coupure de mots ou par les photocomposeuses, dans des programmes spécialisés dans la coupure des mots [14].

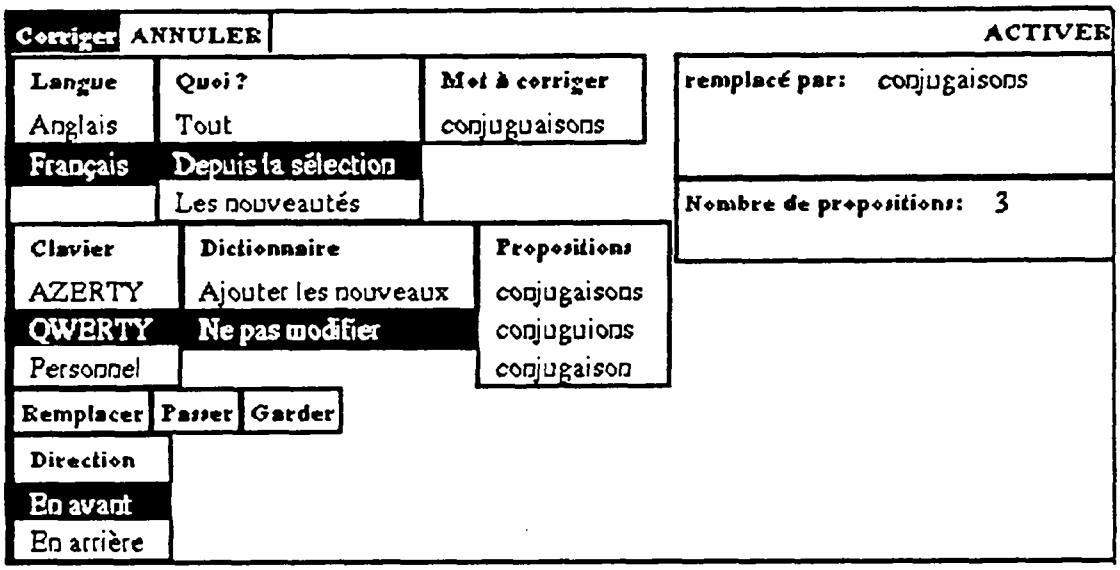
3.3 Interface

Outre le choix d'options de correction, qui permettent d'agir indirectement sur les coefficients décrits en 2.3 et sur le choix des dictionnaires, l'utilisateur a la possibilité de décider de la correction qu'il veut effectuer. En effet, lorsqu'une erreur est détectée par le correcteur, soit il s'agit d'une véritable erreur, soit c'est un mot qui est correct, mais ne figure dans aucun dictionnaire.

S'il s'agit d'une véritable erreur, l'utilisateur peut la corriger immédiatement en utilisant l'une des propositions qui lui sont faites par le correcteur ou par tout autre mot de son choix. Par contre, si l'utilisateur décide que ce mot est correct, il peut immédiatement demander son insertion dans le dictionnaire associé au document de telle sorte que les occurrences de ce mot, rencontrées ultérieurement dans ce document, ne soient pas détectées comme des erreurs : c'est souvent le cas de noms propres ou du vocabulaire spécialisé ou technique. Enfin, s'il s'agit d'une occurrence exceptionnelle, d'un mot qui ne doit pas être rencontré ailleurs dans le document, l'utilisateur peut de-

mander à ce que ce mot soit *marqué* dans le document comme étant un mot *de langue inconnue*. Ainsi, lors des corrections ultérieures de ce document, ce mot ne sera pas pris en compte par le correcteur.

Lorsque Grif a détecté un mot erroné, ce mot apparaît sélectionné (en blanc sur sur fond noir) au milieu de son contexte (avec la partie du document qui le contient) dans la vue principale du document. La correction proposée par le correcteur peut être immédiatement validée en clicquant sur le bouton **Remplacer**. Le correcteur passe alors à la correction du mot erroné suivant.



En résumé, lorsqu'un mot erroné est détecté par le correcteur (c'est le mot courant), l'utilisateur dispose de trois commandes principales :

- **Remplacer** : pour substituer au mot courant erroné le mot de remplacement qu'il a choisi ou modifié lui même,
- **Garder** : pour marquer ce mot courant avec un attribut de *langue inconnue*,
- **Passer** : pour passer au mot erroné suivant, sans rien changer sur le mot erroné courant.

Les options (mise à jour automatique du dictionnaire, langue de correction, nombre de propositions de correction) sont modifiables à tout moment. Les dictionnaires utilisés pour détecter et corriger les erreurs sont les 4 dictionnaires par défaut définis en 3.2. Cependant, si l'utilisateur a préalablement choisi d'autres dictionnaires pour ce document (les noms des dictionnaires sont des *attributs* posés à la racine de chaque document), seuls ces derniers seront utilisés. Ces attributs sont modifiables interactivement comme tous les autres attributs d'un document Grif.

4 Conclusion

Une méthode de correction à base de dictionnaire permet de corriger la plupart des fautes de frappe, mais aussi certaines fautes d'orthographe (phonétiques), avec de bonnes performances (temps de réponse). Dans une population où la connaissance des règles de la grammaire française est "normale", ce type de correction donne de très bons résultats, aussi bien par le taux d'erreurs détectées que par la pertinence des propositions (dans la plupart des cas, la première proposition de correction est la bonne).

Cet algorithme de correction, indépendant de la langue, est utilisable dans un environnement multilingue. Pour l'intégrer dans Grif, nous avons dû prendre en compte la spécificité de deux langues (l'anglais et le français). Mais l'ouverture vers d'autres langues suppose que l'on dispose d'un codage des caractères suffisant pour ces langues et pour les dictionnaires et que l'on sache analyser un texte pour y reconnaître les mots de cette langue. Les développements autour de Grif dans le cadre du projet Opéra se poursuivent dans ce sens, vers une amélioration du contenu textuel des documents.

Références

[1] J. André, R. Furuta, V. Quint, *Structured Documents*, Cambridge University Press, 1989.

[2] J. André et V. Quint, "Structures et modèles de documents" *Le document Electronique*, num. édité par C. Bornes, 1-57, Inria 1990.

[3] B. Van Berkel, K. De Smedt, "Triphone analysis: a combined method for the corrections of orthographical and typographical errors" *Second Conference on Applied Natural Language Processing (Austin)*, 77-83, Février 1988.

[4] R.C. Blair, "A program for correcting spelling errors" *Informat. Contr.*, vol. 3, 60-67, Mars 1960.

[5] *Code typographique*, Fédération nationale du personnel d'encadrement des industries polygraphiques et de la communication, Paris, 13e édition 1981.

[6] F.J. Damerau, "A technique for computer detection and correction of spelling errors" *Communication of the ACM*, vol. 7, num.3, 171-176, Mars 1964.

[7] *FrameMaker, Manuel d'utilisation*, Frame Technology Corporation, San José, California, Mai 1990.

[8] P. Frison, D. Lavenier, "A fast machine for prototyping string correction algorithms" *International Conference on the user-oriented content-based text and image handling, RIAO88*, Cambridge MIT, Mars 1988.

[9] R. Furuta, V. Quint, J. André, "Interactively Editing Structured Documents" *Electronic Publishing - Origination, Dissemination and Design*, vol. 1, num. 1, 19-44, Avril 1988.

[10] *Guide du typographe romand*, le Groupe de Lausanne de l'Association suisse des compositeurs à la machine, Lausanne, Suisse, 1982.

[11] *Lexique des règles typographiques en usage à l'Imprimerie nationale*, Paris, 3e édition 1990.

[12] V.I. Levenshtein, "Binary codes capable of correction deletions, insertion and reversals" *Sov. Phys. Dokl*, vol. 10, 707-710, Février 1966.

[13] B. Marti, all., *Télématique: techniques, normes, services*, Dunod Informatique, Paris, France, 1990.

[14] R. McIntosh, *Hyphenation*, Computer Hyphenation Ltd, Bradford, England, 1990.

[15] J.L. Peterson, "Computer programs for detecting and correcting spelling errors" *Communication of the ACM*, vol. 23, num. 12, 676-687, Décembre 1980.

[16] G. Pérennou, P. Daubeze, F. Lahens, "La vérification et la correction automatique de textes : le système VORTEX" *TSI*, vol. 5, num. 4, 285-305, Juillet 1986.

[17] G. Pérennou, "Les vérificateurs orthographiques" *Texte et ordinateur: les mutations du Lire-Écrire*, Editions de l'espace européen, Actes du colloque interdisciplinaire, Université Paris X, Nanterre, 53-84, Juin 1990.

[18] P. Pujo, G. Langlet, "Aide intelligente pour la correction des requêtes en langage naturel à une base de données" *TSI*, vol. 9, num. 3, 217-230, 1990.

[19] V. Quint, M. Nanard, J. André, "Towards Document Engineering" *EP'90*, R. Furuta, ed., 17-29, Cambridge University Press, Septembre 1990.

[20] V. Quint, I. Vatton, J. André, H. Richy, "Grif et l'édition de documents structurés: nouveaux développements" *Cahiers GUTenberg*, num. 9, 49-65, Juillet 1991.

[21] D. Richard, G. Lapalme, "Un système de correction automatique des accords des participes passés" *TSI*, vol. 5, num. 4, 307-319, Juillet 1990.

[22] J.L. Scharbarg, *Une machine systolique adaptée à la correction des chaînes de caractères. Application aux adresses postales*, Thèse, mention Informatique, Université de Rennes I, Juillet 1990.

[23] J. Véronis, "Phonographic vs typographical correction in natural language interfaces" *Fifth International Synposium in Applied Informatics (Grindenwald, Suisse)*, 180-183, Février 1987.

[24] R.A. Wagner, M.J. Fischer, "The String-to-String Correction Problem" *Journal of ACM*, vol. 21, num. 1, 168-173, Janvier 1974.

LISTE DES DERNIERES PUBLICATIONS INTERNES IRISA

PI 604 A GENERAL METHOD TO DEFINE QUORUMS
Mitchell L. NEILSEN, Masaaki MIZUNO, Michel RAYNAL
Septembre 1991, 20 pages.

PI 605 OBTENTION DES EQUATIONS DYNAMIQUES D'UN SYSTEME PHYSIQUE A PARTIR DE SON MODELE BOND GRAPH
Bénédicte EDIBE
Septembre 1991, 26 pages.

PI 606 CONSTRUCTIVE PROBABILITY AND THE SIGNalea LANGUAGE : BUILDING AND HANDLING RANDOM PROCESSES VIA PROGRAMMING
Albert BENVENISTE
Septembre 1991, 60 pages.

PI 607 ABOUT LOGICAL CLOCKS FOR DISTRIBUTED SYSTEMS
Michel RAYNAL
Octobre 1991, 16 pages.

PI 608 UNE NOUVELLE APPROCHE REALISTE DE SIMULATION D'ECLAIRAGE DANS UN ENVIRONNEMENT DIFFUS
Eric LANGUENOU, Kadi BOUATOUCH, Pierre TELLIER
Octobre 1991, 64 pages.

PI 609 INTEGRATION D'UN CORRECTEUR ORTHOGRAPHIQUE DANS L'EDITEUR STRUCTURE GRIF
Patrice FRISON, Eric PICHERAL, Hélène RICHY
Octobre 1991, 22 pages.

PI 610 SYNCHRONIZATION AND CONCURRENCY MEASURES FOR DISTRIBUTED COMPUTATIONS
Michel RAYNAL
Octobre 1991, 20 pages.

ISSN 0249-6399